

Docket No.: 03226/493002; SUN040039
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Arvind Prabhakar et al.

Application No.: 10/618,035

Confirmation No.: 4145

Filed: June 4, 2004

Art Unit: 2456

For: SYSTEM AND METHOD FOR
TRANSLATING FULLY QUALIFIED
DOMAIN NAME ACCESS IN A BROWSER
ENVIRONMENT

Examiner: K. S. Mai

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on September 23, 2009, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

TABLE OF CONTENTS

I.	Real Party of Interest	4
II.	Related Appeals and Interferences.....	4
III.	Status of Claims	4
IV.	Status of Amendments	5
V.	Summary of Claimed Subject Matter	5
VI.	Grounds of Rejection to be Reviewed on Appeal.....	8
VII.	Argument	8
A.	Examiner has failed to produce a prima facie case of obviousness to reject claims 31-33, 36-39, 42-45, and 48 based on Nielsen and Berstis	8
B.	Examiner has failed to produce a prima facie case of obviousness to reject claims 34, 35, 40, 41, 46, and 47 based on Nielsen, Berstis, and Teare	15
VIII.	Summary	16
	CLAIMS APPENDIX.....	18
	EVIDENCE APPENDIX.....	22
	RELATED PROCEEDINGS APPENDIX	23

TABLE OF AUTHORITIES**Cases**

<i>Graham v. John Deere Co.</i> , 86 S.Ct. 684 (1966).....	8
<i>In Re Geisler</i> , 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed. Cir. 1997).....	14
<i>In re Kahn</i> , 441 F.3d 977, 985-986 (Fed. Cir. 2006).....	16
<i>In re Ratti</i> , 270 F.2d 810, 123 USPQ 349 (CCPA 1959).....	14
<i>In re Robertson</i> , 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999).....	13
<i>In re Rouffet</i> , 149 F.3d 1350, 1355 (Fed. Cir. 1998)	17
<i>In re Wilson</i> , 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)	10
<i>KSR International Co. v. Teleflex Inc.</i> , 127 S.Ct. 1727, 1739, 75 U.S.L.W. 4289 (2007)	8, 9

Statutes

35 U.S.C. §103.....	8, 9, 16
---------------------	----------

Other Authorities

37 C.F.R. § 41.37(c)(1)(vii).....	17
MPEP 2142	15, 16

I. Real Party of Interest

The real party of interest for the referenced application is Sun Microsystems, Inc. An Assignment transferring all interest in the referenced application from the inventors to Sun Microsystems, Inc. was filed with the USPTO on June 4, 2004. The Assignment is recorded at Reel 015421, Frame 0706.

II. Related Appeals and Interferences

To the best of the knowledge of the Appellants and Appellants' legal representative, there are no other appeals or interferences that will directly affect, be affected by, or have a bearing on the decision of the Board of Patent Appeals and Interferences ("the Board") in this appeal.

III. Status of Claims

U.S. Patent Application Serial No. 10/618,035 ("the '035 Application") was filed on June 4, 2004. As filed, the '035 Application included claims 1-30. In an amendment filed July 7, 2008, claims 1-30 were canceled, and claims 31-48 were added. Accordingly, claims 31-48 are pending in the '035 Application. Claims 31, 37, and 43 are independent. The remaining claims depend, either directly or indirectly, from claims 31, 37, and 43. All the pending claims were rejected in an Office Action dated July 27, 2009 ("Office Action"). Claims 31-48 are on appeal.

IV. Status of Amendments

All of the amendments have been entered and considered by the Examiner. No amendments have been filed subsequent to the Office Action. The pending claims of record are presented in the Claims Appendix.

V. Summary of Claimed Subject Matter

The following discussion summarizes the content of the claimed subject matter, and is not intended to limit the scope of the invention. The references to the Specification referenced below should not be construed as the only locations in the Specification which support or discuss the respective limitation.

Independent claim 31 relates to a system for translating domain names. *See, e.g.*, Specification, page 5, lines 5-17; Figure 4. The system includes a Uniform Resource Locator (URL) detection module configured to receive a URL request from a user to access a destination fully qualified domain name (FQDN). *See, e.g.*, Specification, page 10, lines 19-23; page 12, lines 20-21; Figure 3; Figure 4, Step 410. The URL detection module is also configured to determine that the URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, wherein the predefined URL stored in the cookie is specified by the user. *See, e.g.*, Specification, page 10, lines 23-26; page 12, line 21 – page 13, line 2; Figure 4, Step 420. The system also includes a URL redirection module configured to receive the invalid URL request from the URL detection module. *See, e.g.*, Specification, page 10, lines 23-26; Figure 3. The URL redirection module is also configured to redirect the invalid URL request to a FQDN translation

module. *See, e.g.*, Specification, page 11, lines 1-6; Figure 3. The system also includes a FQDN translation module, configured to translate the invalid URL request to a target valid FQDN using a FQDN mapping module, where the FQDN mapping module is stored on a computer readable storage medium. *See, e.g.*, Specification, page 11, lines 8-26; Figure 3; Figure 4, Step 455.

Dependent claim 34 relates to the system of claim 31, where the URL request comprises an alias, and where the alias is stored in the FQDN mapping module. *See, e.g.*, Specification, page 11, line 8 – page 12, line 16; page 13, lines 16-24; Figure 4, Step 450.

Independent claim 37 relates to a method for translating domain names. *See, e.g.*, Specification, page 5, lines 5-17; Figure 4. The method involves receiving, by a Uniform Resource Locator (URL) detection module, a URL request from a user to access a destination fully qualified domain name (FQDN). *See, e.g.*, Specification, page 10, lines 19-23; page 12, lines 20-21; Figure 3; Figure 4, Step 410. The method also involves determining, by the URL detection module that the URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, where the predefined URL stored in the cookie is specified by the user. *See, e.g.*, Specification, page 10, lines 23-26; page 12, line 21 – page 13, line 2; Figure 4, Step 420. The method also involves receiving, by a URL redirection module, the invalid URL request from the URL detection module. *See, e.g.*, Specification, page 10, lines 23-26; Figure 3. The method also involves redirecting, by the URL redirection module, the invalid URL request to a FQDN translation module. *See, e.g.*, Specification, page 11, lines 1-6; Figure 3. The method also involves translating, by the FQDN translation module, the invalid URL request to a target valid FQDN using a FQDN mapping module. *See, e.g.*, Specification, page 11, lines 8-17; Figure 3; Figure 4, Step

455. The method also involves directing the user to a web site associated with the target valid FQDN. *See, e.g.*, Specification, page 11, lines 1-3; Figure 3.

Independent claim 43 relates to a computer readable medium comprising executable instructions for translating domain names. *See, e.g.*, Specification, page 5, lines 5-17; Figure 4. The instructions include receiving, by a Uniform Resource Locator (URL) detection module, a URL request from a user to access a destination fully qualified domain name (FQDN). *See, e.g.*, Specification, page 10, lines 19-23; page 12, lines 20-21; Figure 3; Figure 4, Step 410. The instructions also include determining, by the URL detection module that the URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, where the predefined URL stored in the cookie is specified by the user. *See, e.g.*, Specification, page 10, lines 23-26; page 12, line 21 – page 13, line 2; Figure 4, Step 420. The instructions also include receiving, by a URL redirection module, the invalid URL request from the URL detection module. *See, e.g.*, Specification, page 10, lines 23-26; Figure 3. The instructions also include redirecting, by the URL redirection module, the invalid URL request to a FQDN translation module. *See, e.g.*, Specification, page 11, lines 1-6; Figure 3. The instructions also include translating, by the FQDN translation module, the invalid URL request to a target valid FQDN using a FQDN mapping module. *See, e.g.*, Specification, page 11, lines 8-17; Figure 3; Figure 4, Step 455. The instructions also include directing the user to a web site associated with the target valid FQDN. *See, e.g.*, Specification, page 11, lines 1-3; Figure 3.

VI. Grounds of Rejection to be Reviewed on Appeal

The present Appeal addresses the following grounds of rejection:

- Whether the Examiner has failed to produce a *prima facie* case of obviousness to reject claims 31-33, 36-39, 42-45, and 48 based on US Patent No. 5,892,919 (“Nielsen”) and US Patent No. 6,092,100 (“Berstis”). For purposes of this appeal, claims 31-33, 36-39, 42-45, and 48 stand or fall together.
- Whether the Examiner has failed to produce a *prima facie* case of obviousness to reject claims 34, 35, 40, 41, 46, and 47 based on Nielsen, Berstis, and US Patent No. 6,151,624 (“Teare”). For purposes of this appeal, claims 34, 35, 40, 41, 46, and 47 stand or fall together.

VII. Argument

A. Examiner has failed to produce a *prima facie* case of obviousness to reject claims 31-33, 36-39, 42-45, and 48 based on Nielsen and Berstis

In this Appeal, Appellants argue that the Examiner has failed to produce a *prima facie* case of obviousness to reject claims 31-33, 36-39, 42-45, and 48 based on Nielsen and Berstis, whether viewed separately or in combination, for at least the reasons given below. Independent claim 31 is representative of claims 31-33, 36-39, 42-45, and 48.

35 U.S.C. §103 provides the statutory definition of obviousness. The framework for applying 35 U.S.C. §103 was initially set out by the Supreme Court in *Graham v. John Deere Co.*, 86 S.Ct. 684 (1966). This framework was reaffirmed by the court in *KSR International Co. v. Teleflex Inc.*, 127 S.Ct. 1727, 1739, 75 U.S.L.W. 4289 (2007). Based on the above framework, one

rationale that may be used to support a conclusion of obviousness is that all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination yielded nothing more than predictable results to one of ordinary skill in the art. *See KSR*, 127 S.Ct. at 1739. In the instant case, the Examiner, in articulating the analysis used to reject the claims under 35 U.S.C. §103, has used the above rationale to support a rejection of obviousness in view of Nielsen and Berstis. *See Office Action*, pp. 3-14.

Appellants submit that the Examiner has incorrectly characterized the cited references. As such, the Examiner has failed satisfy the requirements of the aforementioned rationale (namely, that all elements were known in the prior art) to establish a *prima facie* case of obviousness of the present claims. Further, Appellants submit that combination proposed by the Examiner is improper, as it would alter the principle of operation of at least one prior art reference and/or the prior art references teach away for the purposed combination. For at least these reasons, the Examiner's rejection should be reversed.

1. The Examiner has incorrectly argued that Nielsen and Berstis disclose a predefined URL

Independent claim 31 requires, in part, *a URL detection module configured to determine that a URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, wherein the predefined URL stored in the cookie is specified by the user*. In other words, claim 31 requires, in part,: (i) determining that the URL request is invalid if the URL request does not match the predefined URL stored in the cookie, and (ii) that the predefined URL stored in the cookie is specified by the user.

In rejecting claim 31, the Examiner relies on a combination of Nielsen and Berstis. *See* Office Action, pp. 4-6. Specifically, the Examiner contends that Figure 5 of Nielsen discloses the claimed *URL detection module*, but admits that Nielsen does not explicitly disclose requirements (i) and (ii) listed above. *See Id.* Instead, the Examiner cites column 5 line 50 to column 6 line 16 and Figure 4 of Berstis to disclose requirements (i) and (ii). *See Id.* The cited portion of Berstis describes a process where “the user enters a character string,” and then “[a]t step 52, a test is done to determine whether the string is recognized.” *See* Berstis, column 5, lines 52-54. Next, if the string is not recognized, “a test is done at step 55 to determine whether the IP address portion of the URL is recognized and a connection established.” *See* Berstis, column 5, lines 56-63. Finally, if the IP address is not recognized, “the character string is indexed into a lexicon of server IP names that have been used by the Web client over a given ‘history’ period.” *See* Berstis, column 6, lines 1-5.

Appellants respectfully disagree with the Examiner’s contentions as to the teachings of Bertis. Specifically, the Examiner has ignored explicit limitations required by the claims, and is thus failing to consider “[a]ll words in a claim.” *See In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). As stated above, claim 31 explicitly requires a predefined URL stored in a cookie. However, the cited portion of Berstis is silent with regard to a predefined URL stored in a cookie. In fact, a review of Berstis reveals that Berstis is entirely silent with regard to storing any information in cookies. Further, a review of Nielsen reveals that Nielsen is also silent with regard to a predefined URL stored in a cookie. Therefore, Nielsen and Berstis, either alone or in combination, fail to disclose at least the aforementioned limitations.

Moreover, even assuming, *arguendo*, that Berstis discloses a predefined URL stored in a cookie, Berstis fails to disclose determining that the URL request is invalid if the URL request does

not match the predefined URL stored in the cookie (*i.e.*, requirement (*i*) listed above). Regarding this requirement, the Examiner contends “Berstis discloses at step 52 a test is done to determine whether the string entered in the address field (URL) is recognized,” and “[i]f the string is recognized (valid), then the client is connected to the target URL.” *See* Office Action, p. 6. Thus, the Examiner contends that determining that the string is not recognized (*i.e.*, step 52 of Berstis) discloses determining that the URL request is invalid, as required by claim 31. However, Berstis is entirely silent with regard to how the test at step 52 is performed. Thus, because Berstis fails to explain how to determine that the string is not recognized (*i.e.*, invalid), Berstis cannot possibly disclose determining that the URL request is invalid when the URL request does not match a predefined URL stored in a cookie (*i.e.*, requirement (*i*) listed above).

Further, a review of Nielsen confirms that, as admitted by the Examiner, Nielsen fails to disclose the aforementioned requirement. *See* Office Action, p. 5. Instead, Nielsen discloses determining whether an issued URL is found in a spell check cache, and if so, then determining that the issued URL is incorrect. *See* Nielsen, Figure 5, step 505. In other words, Nielsen teaches storing incorrect URLs in the spell check cache, and determining that an issued URL is incorrect when it matches the spell check cache. In contrast, the present claims require an entirely different approach, namely storing correct URLs in cookies, and determining that a URL request is incorrect when it does not match the correct URLs stored in the cookies. Thus, Nielsen also fails to disclose requirement (*i*) listed above. Accordingly, Nielsen and Berstis, either alone or in combination, fail to disclose at least requirement (*i*) listed above

Moreover, even assuming, *arguendo*, that Berstis discloses requirement (*i*), Appellants submit that Berstis fails to disclose that the predefined URL stored in the cookie is specified by the

user (*i.e.*, requirement (ii) listed above). Regarding this requirement, the Examiner asserts “[i]t is seen that the list of recognized URLs were specified by the user because it is a list a previously contacted URLs, and in order for the URLs to be previously contacted they must have been originally been specified by the user in an attempt to access those URLs.” *See* Office Action, p. 6. However, as explained above, Berstis is entirely silent regarding how to determine whether the URL string is recognized at step 52. Thus, contrary to the Examiner’s contentions, Berstis does not disclose any list of URLs used at step 52 to determine whether the string is recognized. Rather, it appears that the list referred to by the Examiner is the “lexicon of server IP names” described in Berstis, column 6, lines 1-5.

Appellants respectfully submit that the Examiner has mischaracterized the “lexicon” of Berstis. Specifically, contrary to the Examiner’s assertions, Berstis fails to disclose that the “lexicon” is used to determine whether a URL string is recognized. Rather, Berstis discloses that, if a string is not recognized (at step 52 of Figure 4), and if the IP address portion of the URL is also not recognized (at step 55 of Figure 4), then the “lexicon” is used to perform a “fuzzy search” to generate “a list of URLs that most closely match what was originally entered into the browser address field,” such that “[t]he user can then select the correct URL from the list.” *See* Berstis, Abstract and Figures 5-6. In other words, the “lexicon” is only used after it has been determined at step 52 that the URL is not recognized, and is used to generate a list of candidates which are most similar to the desired URL. Accordingly, the Examiner’s rejection is based on an incorrect interpretation of Berstis.

Moreover, even assuming, *arguendo*, that the “lexicon” discloses a predefined URL used to determine whether a user-entered URL is valid, Appellants submit that the “lexicon” is not specified

by the user. Specifically, the Examiner contends that the “lexicon” is a list of previously contacted URLs, and that “in order for the URLs to be previously contacted they must have originally been specified by the user in an attempt to access those URLs.” *See* Office Action, p. 6. In other words, the Examiner contends that the requirement that the URLs in the “lexicon” are specified by the user is inherent because the “lexicon” includes “server IP names” for websites visited by the user.

Appellants disagree with the Examiner’s contentions. In order to establish inherency, the Examiner must show that “the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.” *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted). Thus, in this case, the Examiner must show that the requirement that a URL in the “lexicon” is specified by the user is made necessarily present by the mere fact that the “lexicon” includes “server IP names that have been used by the Web client.”

However, one of skill in the art will appreciate that a Web client may display a website without the URL of that website having been specified by a user. For example, a user clicking on a hyperlinked object embedded in a first website may cause a second website to be displayed. In another example, the display of a first website may cause a second website to be displayed automatically as a “popup.” In yet another example, the URL of a first website may be automatically redirected to a second website via domain forwarding. In yet another example, a website may be displayed as a random selection or as part of search results (*e.g.*, by using the “I’m Feeling Lucky” button on the Google search page). In all of these examples, the “lexicon” of Berstis would include the “server IP name” of the visited website, without the URL of the second website ever being specified by the user. Therefore, a predefined URL specified by the user is not

necessarily present in the “lexicon,” and therefore is clearly not inherent in Berstis. Thus, it follows logically that Berstis cannot disclose or render obvious that the predefined URL being specified by the user, as required by claim 31. Further, a review of Nielsen reveals that Nielsen is also silent with regard to the predefined URL being specified by the user. Accordingly, Nielsen and Berstis, either alone or in combination, fail to disclose at least requirement (ii) listed above.

2. The combination of prior art references proposed by the Examiner is improper

Appellants submit that the combination of references proposed by the Examiner is improper, and thus fails to establish a *prima facie* case of obviousness. Specifically, the courts have found that, when combining prior art elements to establish obviousness, if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *See In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Additionally, under *in re Geisler*, “[a] prima facie case of obviousness may also be rebutted by showing that the art, in any material respect, teaches away from the claimed invention.” *See In Re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed. Cir. 1997) *See In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

In the present case, the modification proposed by the Examiner would completely alter the principle of operation of Nielsen, namely the use of cached misspellings to detect spelling errors. Moreover, Nielsen not only discloses the use of cached misspellings to detect spelling errors, but also discloses the importance of that principle of operation. Specifically, Nielsen discloses that “a cache of recent URL misspellings provides a quick way to trap spelling errors” and “cached spell

checking will minimize user inconvenience by transparently correcting the URL.” *See* Nielsen, col. 2, lines 51-59. Accordingly, by emphasizing the importance of the use of cached misspellings, Nielsen actually teaches away from the modification proposed by the Examiner (*i.e.*, storing correct URLs in a cookie instead of storing cached misspellings). Thus, for at least the above reasons, the modification proposed by the Examiner is improper, and thus fails to establish a *prima facie* case of obviousness.

3. Summary

Based on the above, the Examiner has failed to show sufficient evidence of *prima facie* obviousness to reject claim 31 in view of Nielsen and Berstis. Moreover, because the Examiner has failed to produce a *prima facie* case of obviousness, the Appellants are under no obligation to submit evidence of nonobviousness. *See* MPEP 2142.

B. Examiner has failed to produce a prima facie case of obviousness to reject claims 34, 35, 40, 41, 46, and 47 based on Nielsen, Berstis, and Teare

In this Appeal, Appellants argue that the Examiner has failed to produce a *prima facie* case of obviousness to reject claims 34, 35, 40, 41, 46, and 47 based on Nielsen, Berstis, and Teare, whether viewed separately or in combination, for at least the reasons given below. Dependent claim 34 is representative of claims 34, 35, 40, 41, 46, and 47.

In rejecting dependent claim 34, the Examiner relies on Nielsen and Berstis as applied to independent claims 31, 37, and 43. *See* Office Action, pp. 14-16. However, as described above, Appellants respectfully submit that Nielsen and Berstis fail to disclose or render obvious at least the

aforementioned limitations of claim 31 (*i.e.*, requirements (i) and (ii) listed above). Thus, Appellants submit that dependent claim 34 is also patentable over Nielsen and Berstis for at least the same reasons as independent claim 31.

Further, Teare fails to supply that which Nielsen and Berstis lack. Specifically, Teare is silent with regard to (i) determining that the URL request is invalid if the URL request does not match the predefined URL stored in the cookie, and (ii) that the predefined URL stored in the cookie is specified by the user, as required by independent claim 31. Accordingly, independent claim 31 is also patentable over Nielsen, Berstis, and Teare. Claim 34 depends from independent claim 31. Accordingly, claim 34 is patentable over Nielsen, Berstis, and Teare for at least the same reasons.

Based on the above, Appellants submit that the Examiner has failed to show sufficient evidence of *prima facie* obviousness to reject claim 34 in view of Nielsen, Berstis, and Teare. Moreover, because the Examiner has failed to produce a *prima facie* case of obviousness, the Appellants are under no obligation to submit evidence of nonobviousness. *See* MPEP 2142.

VIII. Summary

In view of the above, as the Examiner has failed to show sufficient evidence for a *prima facie* case of obviousness, the Appellants have carried their burden in showing that the Examiner erred in finally rejecting claims 31-48 under 35 U.S.C. §103. *In re Kahn*, 441 F.3d 977, 985-986 (Fed. Cir. 2006) (“On appeal to the Board, an applicant can overcome a rejection by showing insufficient evidence of *prima facie* obviousness or by rebutting the *prima facie* case with evidence

of secondary indicia of nonobviousness”) (emphasis in original) (quoting *In re Rouffèt*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)); *see also* 37 C.F.R. § 41.37(c)(1)(vii). Favorable consideration of the present application is respectfully requested.

Dated: November 23, 2009

Respectfully submitted,

By /Robert P. Lord/
Robert P. Lord
Registration No.: 46,479
OSHA · LIANG LLP
3945 Freedom Circle, Suite 300
Santa Clara, California 95054
(408) 727-0600
(713) 228-8778 (Fax)
Attorney for Appellants

CLAIMS APPENDIX

Claims Involved in Appeal.

1. – 30. (Cancelled)

31. A system for translating domain names comprising:
- a Uniform Resource Locator (URL) detection module, configured to:
 - receive a URL request by a user to access a destination fully qualified domain name (FQDN), and
 - determine that the URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, wherein the predefined URL stored in the cookie is specified by the user;
 - a URL redirection module, configured to:
 - receive the invalid URL request from the URL detection module, and
 - redirect the invalid URL request to a FQDN translation module; and
 - the FQDN translation module, configured to:
 - translate the invalid URL request to a target valid FQDN using a FQDN mapping module, wherein the FQDN mapping module is stored on a computer readable storage medium.
32. The system of claim 31, further comprising:
- a FQDN default setter configured to provide a predefined default target valid FQDN, wherein the FQDN default setter is used by the FQDN mapping module.
33. The system of claim 31, wherein the FQDN mapping module is configured to provide a mapping between the invalid URL request and the target valid FQDN.
34. The system of claim 31, wherein the URL request comprises an alias, wherein the alias is stored in the FQDN mapping module.

35. The system of claim 34, wherein the FQDN mapping module comprises a mapping of the alias to the target valid FQDN.
36. The system of claim 31, wherein the URL detection module, the URL redirection module, and the FQDN translation module execute in a browser.
37. A method for translating domain names, comprising:
receiving, by a Uniform Resource Locator (URL) detection module, a URL request from a user to access a destination fully qualified domain name (FQDN), and
determining, by the URL detection module that the URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, wherein the predefined URL stored in the cookie is specified by the user;
receiving, by a URL redirection module, the invalid URL request from the URL detection module;
redirecting, by the URL redirection module, the invalid URL request to a FQDN translation module;
translating, by the FQDN translation module, the invalid URL request to a target valid FQDN using a FQDN mapping module; and
directing the user to a web site associated with the target valid FQDN.
38. The method of claim 37, further comprising:
providing a predefined default target valid FQDN by a FQDN default setter, wherein the FQDN default setter is used by the FQDN mapping module.
39. The method of claim 37, wherein the FQDN mapping module is configured to provide a mapping between the invalid URL request and the target valid FQDN.
40. The method of claim 37, wherein the URL request comprises an alias, wherein the alias is stored in the FQDN mapping module.
41. The method of claim 40, wherein the FQDN mapping module comprises a mapping of the alias to the target valid FQDN.

42. The method of claim 37, wherein the URL detection module, the URL redirection module, and the FQDN translation module execute in a browser.
43. A computer readable medium comprising executable instructions for translating domain names by:
- receiving, by a Uniform Resource Locator (URL) detection module, a URL request from a user to access a destination fully qualified domain name (FQDN), and
 - determining, by the URL detection module that the URL request is an invalid URL request when the URL request is inconsistent with a predefined URL stored in a cookie, wherein the predefined URL stored in the cookie is specified by the user;
 - receiving, by a URL redirection module, the invalid URL request from the URL detection module;
 - redirecting, by the URL redirection module, the invalid URL request to a FQDN translation module;
 - translating, by the FQDN translation module, the invalid URL request to a target valid FQDN using a FQDN mapping module; and
 - directing the user to a web site associated with the target valid FQDN.
44. The computer readable medium of claim 43, further comprising:
- providing a predefined default target valid FQDN by a FQDN default setter, wherein the FQDN default setter is used by the FQDN mapping module.
45. The computer readable medium of claim 43, wherein the FQDN mapping module is configured to provide a mapping between the invalid URL request and the target valid FQDN.
46. The computer readable medium of claim 43, wherein the URL request comprises an alias, wherein the alias is stored in the FQDN mapping module.
47. The computer readable medium of claim 46, wherein the FQDN mapping module comprises a mapping of the alias URL request to the target valid FQDN.

48. The computer readable medium of claim 43, wherein the URL detection module, the URL redirection module, and the FQDN translation module execute in a browser.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE